REPORT DOCUMENTATION PAGE	Form Approved OMB NO. 0704-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggesstions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA, 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any oenalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE (DD-MM-YYYY) 03-10-2016	2. REPORT TYPE Final Report	3. DATES COVERED (From - To) 31-May-2013 - 30-May-2016

4. TITLE AND SUBTITLE Final Report: MCloud: Secure Provenance for Mobile Cloud Users	5a. CONTRACT NUMBER W911NF-13-1-0142
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER 206022
6. AUTHORS Bogdan Carbunar	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES Florida International University 10555 West Flagler, EC 2441 Miami, FL 33174 -1630	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS (ES) U.S. Army Research Office P.O. Box 12211 Research Triangle Park, NC 27709-2211	10. SPONSOR/MONITOR'S ACRONYM(S) ARO
	11. SPONSOR/MONITOR'S REPORT NUMBER(S) 62844-CS-REP.25

12. DISTRIBUTION AVAILIBILITY STATEMENT

Approved for Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES

The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT

In this project we have developed an array of solutions for protecting and detecting fraudulent mobile device and social network data. At device level, we designed KXRay to detect the existence and location of specific instances of target data structure types in kernel or VM by observing memory accesses and training for target-specific timing-based signatures. We have developed DroidShield, a new Android/TrustZone protection paradigm that protects user-land application data from all unauthorized accesses. We have built FitBite and Garmax, tools that attack the storage and communication protocols of sensor based trackers Fitbit and Garmin; we have developed SensCrypt, a

15. SUBJECT TERMS

fraud detection, data protection, provenance, mobile device, social network

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON Bogdan Carbunar
a. REPORT UU	b. ABSTRACT UU	c. THIS PAGE UU	UU		19b. TELEPHONE NUMBER 305-348-7566

Report Title

Final Report: MCloud: Secure Provenance for Mobile Cloud Users

ABSTRACT

In this project we have developed an array of solutions for protecting and detecting fraudulent mobile device and social network data. At device level, we designed KXRay to detect the existence and location of specific instances of target data structure types in kernel or VM by observing memory accesses and training for target-specific timing-based signatures. We have developed DroidShield, a new Android/TrustZone protection paradigm that protects user-land application data from all unauthorized accesses. We have built FitBite and Garmax, tools that attack the storage and communication protocols of sensor based trackers Fitbit and Garmin; we have developed SensCrypt, a computation and storage efficient solution for securing the storage and communication of resource constrained trackers. In addition, at mobile application level, we have introduced Marco and Vamos, systems that detect plagiarized videos, falsely claimed to have been captured on mobile devices.

At social network level, we have introduced FairPlay and Marco, systems that detect search rank fraud in Google Play and Yelp, respectively. We have designed GeoPal, a mobile app that enables users to detect and defend against friend spam in Facebook.

Enter List of papers submitted or published that acknowledge ARO support from the start of the project to the date of this printing. List the papers, including journal references, in the following categories:

(a) Papers published in peer-reviewed journals (N/A for none)

Received		Paper
07/25/2015	12.00	Jaime Ballesteros, Bogdan Carbunar, Mahmudur Rahman, Naphtali Rishe, S. S. Iyengar. Towards Safe Cities: A Mobile and Social Networking Approach, IEEE Transactions on Parallel and Distributed Systems, (09 2014): 2451. doi: 10.1109/TPDS.2013.190
07/25/2015	13.00	Bogdan Carbunar, Jaime Ballesteros, Duen Horng Polo Chau, Mahmudur Rahman. To catch a fake: Curbing deceptive Yelp ratings and venues, Statistical Analysis and Data Mining, (04 2015): 147. doi: 10.1002/sam.11264
07/28/2015	17.00	Bhushan Jain, Mirza Basim Baig, Dongli Zhang, Donald E. Porter, Radu Sion. Introspections on the Semantic Gap, IEEE Security & Privacy, (03 2015): 48. doi: 10.1109/MSP.2015.35
08/12/2014	4.00	Bogdan Carbunar, Mahmudur Rahman, Niki Pissinou, Athanasios Vasilakos. A survey of privacy vulnerabilities and defenses in geosocial networks, IEEE Communications Magazine, (11 2013): 114. doi: 10.1109/MCOM.2013.6658662
08/12/2014	6.00	Bogdan Carbunar, Mahmudur Rahman, Jaime Ballesteros, Naphtali Rishe, Athanasios V. Vasilakos. ProfilR: Toward Preserving Privacy and Functionality in Geosocial Networks, IEEE Transactions on Information Forensics and Security, (04 2014): 709. doi: 10.1109/TIFS.2014.2307697

TOTAL: **5**

Number of Papers published in peer-reviewed journals:

(b) Papers published in non-peer-reviewed journals (N/A for none)

Received		Paper
08/12/2014	1.00	Bogdan Carbunar, Radu Sion, Rahul Potharaju, Moussa Ehsan. Private Badges for GeoSocial Networks, , ( ): 0. doi:
TOTAL:		**1**

Number of Papers published in non peer-reviewed journals:

(c) Presentations

Number of Presentations: 0.00

Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received		Paper
09/06/2016	21.00	. A Longitudinal Study of the Google App Market, the 2015 IEEE/ACM ASONAM. 25-AUG-15, Paris, France. : ,
09/06/2016	20.00	. Yelp Events: Making Bricks Without Clay?, 2013 IEEE 33rd International Conference on Distributed Computing Systems Workshops (ICDCSW). 08-JUL-13, Philadelphia, PA, USA. : ,
09/06/2016	22.00	. FairPlay: Fraud and Malware Detection in Google Play, Proceedings of the 2016 SIAM International Conference on Data Mining. 06-MAY-16, Miami. : ,
09/06/2016	23.00	. GeoPal: Friend Spam Detection in Social Networks Using Private Location Proofs, IEEE International Conference on Sensing, Communication and Networking (SECON). 27-JUN-16, London. : ,
09/06/2016	24.00	. DroidShield: Protecting User Applications from Normal, ACM CCS. 06-OCT-16, Viena. : ,
TOTAL:		**5**

Number of Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received		Paper
09/06/2016	2.00	Mahmudur Rahman, Umut Topkara, Bogdan Carbunar. Seeing is Not Believing: Visual Verifications Through Liveness Analysis using Mobile Devices, the 29th Annual Computer Security Applications Conference. 09-DEC-13, New Orleans, Louisiana. : ,
09/06/2016	7.00	Bhushan Jain, Mirza Basim Baig, Dongli Zhang, Donald E. Porter, Radu Sion. SoK: Introspections on Trust and the Semantic Gap, Proceedings of the 2014 IEEE Symposium on Security and Privacy. 18-MAY-14, Oakland. : ,
09/06/2016	8.00	Mirza Basim Baig, Connor Fitzsimons, Suryanarayanan Balasubramanian, Radu Sion, Donald E. Porter. CloudFlow: Cloud-wide policy enforcement using fast VM introspection, IC2E '14 Proceedings of the 2014 IEEE International Conference on Cloud Engineering. 10-MAR-14, Boston. : ,
09/06/2016	16.00	Moussa Ehsan, Radu Sion, Chen Chen. Quantitative Musings on the Feasibility of Smartphone Clouds, 2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid). 04-MAY-15, Shenzhen, China. : ,
09/06/2016	11.00	Mahmudur Rahman, Bogdan Carbunar, Umut Topkara. SensCrypt: A Secure Protocol for Managing Low Power Fitness Trackers, 2014 IEEE 22nd International Conference on Network Protocols (ICNP). 21-OCT-14, Raleigh, NC, USA. : ,
09/06/2016	10.00	Anita Wu, Sebastian Ramirez, Ian Michael Terry, Alex Pissinou Makki, Leonardo Bobadilla, Niki Pissinou, S.S. Iyengar, Bogdan Carbunar. Geofit: Verifiable Fitness Challenges, 2014 IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems (MASS). 28-OCT-14, Philadelphia, PA, USA. : ,
09/06/2016	14.00	Mahmudur Rahman, Mozhgan Azimpourkivi, Umut Topkara, Bogdan Carbunar. Liveness verifications for citizen journalism videos, the 8th ACM Conference on Security and Privacy in Wireless and Mobile Networks. 22-JUN-15, New York, New York. : ,
TOTAL:		7

Number of Peer-Reviewed Conference Proceeding publications (other than abstracts):

(d) Manuscripts

Received	Paper

TOTAL:

Number of Manuscripts:

Books

Received	Book

TOTAL:

Received	Book Chapter

TOTAL:

Patents Submitted

Patents Awarded

Awards

Nicholas D. Georganas Best Paper Award 2014, from the ACM Transactions on Multimedia Computing Communications and Applications (TOMCCAP).

Best student paper award for the paper "Turning the Tide: Curbing Deceptive Yelp Behaviors" that appeared in SIAM SDM 2014.

Graduate Students

NAME	PERCENT_SUPPORTED	Discipline
Mahmudur Rahman	1.00	
Mozhgan Azimpurkivi	0.50	
Mirza Basim Baig	0.75	
Sumeet Bajaj	0.37	
Chen Chen	0.38	
FTE Equivalent:	**3.00**	
Total Number:	**5**	

Names of Post Doctorates

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Names of Faculty Supported

NAME	PERCENT_SUPPORTED	National Academy Member
Bogdan Carbunar	0.33	
Radu Sion	0.00	
FTE Equivalent:	**0.33**	
Total Number:	**2**	

Names of Under Graduate students supported

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Student Metrics

This section only applies to graduating undergraduates supported by this agreement in this reporting period

The number of undergraduates funded by this agreement who graduated during this period: 0.00

The number of undergraduates funded by this agreement who graduated during this period with a degree in science, mathematics, engineering, or technology fields:...... 0.00

The number of undergraduates funded by your agreement who graduated during this period and will continue to pursue a graduate or Ph.D. degree in science, mathematics, engineering, or technology fields:...... 0.00

Number of graduating undergraduates who achieved a 3.5 GPA to 4.0 (4.0 max scale):...... 0.00

Number of graduating undergraduates funded by a DoD funded Center of Excellence grant for Education, Research and Engineering:...... 0.00

The number of undergraduates funded by your agreement who graduated during this period and intend to work for the Department of Defense 0.00

The number of undergraduates funded by your agreement who graduated during this period and will receive scholarships or fellowships for further studies in science, mathematics, engineering or technology fields:...... 0.00

Names of Personnel receiving masters degrees

NAME

Total Number:

Names of personnel receiving PHDs

NAME	
Mahmudur Rahman	
Total Number:	**1**

Names of other research staff

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Sub Contractors (DD882)

1 a. Stony Brook University

1 b. 100 Nicolls Rd
Stony Brook NY 11794

Sub Contractor Numbers (c): 8048782470000

Patent Clause Number (d-1):

Patent Date (d-2):

Work Description (e):

Sub Contract Award Date (f-1): 6/1/13 12:00AM

Sub Contract Est Completion Date(f-2): 5/31/14 12:00AM

1 a. Stony Brook University

1 b. 100 Nicolls Rd
Stony Brook NY 11794

Sub Contractor Numbers (c): 8048782470000

Patent Clause Number (d-1):

Patent Date (d-2):

Work Description (e):

Sub Contract Award Date (f-1): 6/1/13 12:00AM

Sub Contract Est Completion Date(f-2): 5/31/14 12:00AM

Inventions (DD882)

Scientific Progress

During the last year of this project, we have made progress on the following areas.

1. Search rank fraud detection in Google Play. We have performed a detailed temporal analysis of Google Play, Google's app market, on data that we collected daily from 160,000 apps, over a period of six months in 2012. We have discovered that at most 50% of the apps are updated in all categories, which significantly impacts the median price. The average price does not exhibit seasonal monthly trends and a changing price does not show any observable correlation with the download count. We have also shown that productive developers are not creating many popular apps, but a few developers control apps which dominate the total number of downloads. In addition, we have collected longitudinal app data from 87,000 apps, 2.9 million reviews, and 2.4 million reviewers, over half a year, between 2014 and 2015. We have developed FairPlay, a novel system that uncovers both malware and search rank fraud apps, by picking out trails that fraudsters leave behind. To identify suspicious apps, FairPlay's PCF algorithm correlates review activities and uniquely combines detected review relation with linguistic and behavioral signals gleaned from longitudinal Google Play app data. We have shown that 75% of the identified malware apps engage in search rank fraud. FairPlay discovers hundreds of fraudulent apps that currently evade Google Bouncer's detection technology, and reveals a new type of attack campaign, where users are harassed into writing positive reviews, and install and review other apps.

2. Friend spam detection. We performed a user study on 68 participants, and discovered that they tend to trust more the Facebook friends with whom they meet more frequently. We have leveraged this result to introduce and build GeoPal, a framework that carefully accesses the potentially sensitive location history of users to privately prove their past location claims, and to privately compute and update fuzzy co-location affinities with other users. We have shown that GeoPal is practical: a Nexus 5 can process more thank 20K location proofs per second.

3. We have built DroidShield, a system that provides a new Android/TrustZone protection paradigm that enables MCloud to protect user-land application data from all unauthorized accesses, even those originating from a compromised kernel, with the highest privilege. MCloud context data gathered by smartphone sensors can now be relayed correctly and with integrity to its intended trusted MCloud code.

Technology Transfer

MCloud: Secure Provenance for Mobile Cloud Users
Final Report

Bogdan Carbunar
Florida International University
Computing and Information Science
Miami, FL, 33199
carbunar@cs.fiu.edu
(305) 851-2436

Radu Sion
Stony Brook University
Computer Science Department
Stony Brook, NY 11794-4400
sion@cs.stonybrook.edu
(631) 731-1695

Social networks and media have transformed mobile device users into human sensors that report data from remote, possibly hard to access areas of interest. For instance, the recent emergence of video sharing sites, e.g., YouTube, Vine, has paved the way toward citizen journalism: people that witness events of public importance (e.g., conflicts, protests, disasters) are now able to post their records of the events and share them with the community at large. In addition, opinions posted on social networks are central to numerous aspects of people's daily online and physical activities. Yet, in critical settings it is especially difficult to ascertain and assert an acceptable level of trust, and current technologies allow easy forging, manipulation and fabrication.

In this project we have developed solutions to establish the authenticity and integrity of social media created on mobile devices, as well as to secure the storage and communications of the mobile devices. This effort is of paramount importance to enable the use of such media for evidence and intelligence gathering purposes. In the following, we describe our results on each dimension studied.

1. Mobile Video Fraud Detection

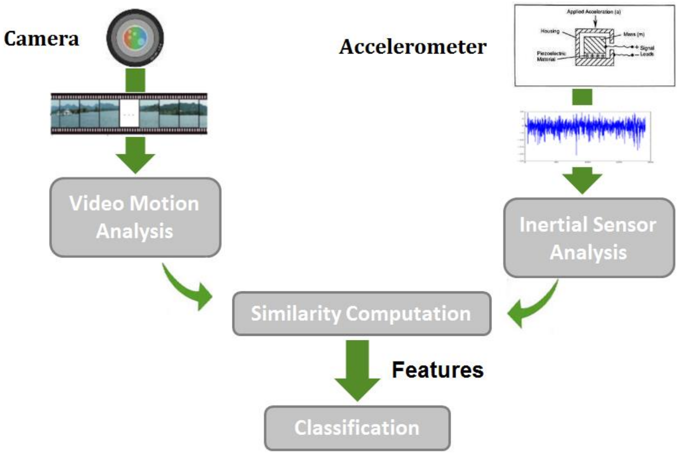
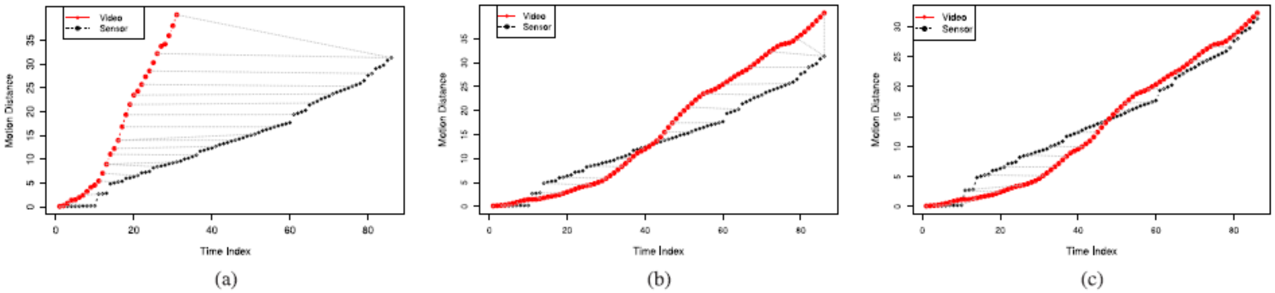


Figure 1. Movee uses four modules to verify a video stream: the i) Video motion analysis, and the ii) Inertial sensor motion analysis, produce movement estimations during capture, iii) Similarity computation extracts features, which iv) classification uses to make the final decision.

We have focused on detecting plagiarized mobile videos. For instance, let us consider a scenario where a malicious party physically present in the U.S. uses a mobile device to ``capture" a video of a projection showing violence previously shot on a different continent. Thus, in addition to assessing the device, location and time of capture, of crucial interest is the ``liveness" dimension of the problem: verify that data has indeed been captured live on a mobile device, and has not been fabricated, e.g., using material from other sources. In the following, we describe Movee and Vamos, two systems we have developed in order to efficiently and securely identify plagiarized videos.

1.1. Movee

We have developed Movee, a first system that addresses the fundamental question of whether the visual stream uploaded by a user has been captured live on a mobile device, and has not been tampered with by an adversary. Movee, illustrated in Figure 1, leverages the mobile device motion sensors and the intrinsic user movements during the shooting of the video. Movee exploits the observation that the movement of the scene recorded on the video stream should be related to the movement of the device simultaneously captured by the accelerometer. Contrary to existing algorithms, Movee has the unique strength of not depending on the audio track.



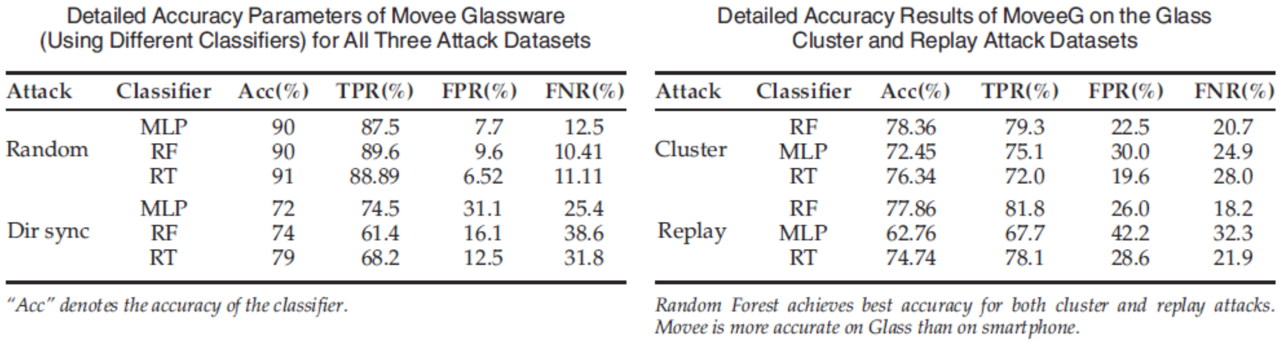
(a) (b) (c)

Figure 2. Example alignment of video and inertial motion streams extracted from the same experiment: (a) When using only DTW. (b) When stretching the shorter vector and applying DTW. (c) After stretching and calibration and applying DTW. Stretching helps achieve a significant alignment improvement.

Movee consists of four modules, illustrated in Figure 1. The video motion analysis (VMA) module processes the video stream captured by the camera. It uses video processing techniques to infer the motion of the camera, producing a time-dependent motion vector. VMA is inspired by the process used in image stabilization capable cameras. The Inertial sensor motion analysis (IMA) module converts the data signal captured from the inertial sensor into another time dependent motion vector. The motion vectors produced by the VMA and IMA modules are compared in the similarity computation (SC) module. SC relies on a flavor of the dynamic time warping (DTW) algorithm to compute the "similarity" of the two motion vectors. It also leverages stretching and calibration techniques in order to address the different frequency of capturing video and acceleration sensors and the difference in motion pattern speed inference based on the subject's distance to the camera. Figure 2 illustrates the effects of DTW, stretching and calibration steps on the alignment of a video and acceleration sample. The SC module also produces a set of features which summarize the nature of the similarity. The features are then used by the classification module, which runs trained classifiers to decide whether the two motion sequences corroborate each other. If they do, Movee concludes the video is genuine.

We have implemented a Movee client using Android and a server component using C++, R and PHP. We used the Open Source Computer Vision (OpenCV) library [7] for the video motion analysis. The client allows users to capture movies and simultaneously provide proofs of liveness. We have also implemented MoveeG, a Movee app variant for the Google Glass, see Fig. 8. We have used the glass development kit (GDK) to build MoveeG as a glassware that runs directly on Glass (around 700 lines of code). MoveeG starts and stops by voice command or through a tap based menu. Since the built-in camera activity has limited functionality, we have built our own logic with the Android Camera API

[2], to capture videos. Once the video capture is completed, MoveeG sends the captured video and accelerometer streams to a server over the Glass Wi-Fi connectivity, using HTTP POST requests.



Detailed Accuracy Parameters of Movee Glassware (Using Different Classifiers) for All Three Attack Datasets

Attack	Classifier	Acc(%)	TPR(%)	FPR(%)	FNR(%)
Random	MLP	90	87.5	7.7	12.5
	RF	90	89.6	9.6	10.41
	RT	91	88.89	6.52	11.11
Dir sync	MLP	72	74.5	31.1	25.4
	RF	74	61.4	16.1	38.6
	RT	79	68.2	12.5	31.8

"Acc" denotes the accuracy of the classifier.

Detailed Accuracy Results of MoveeG on the Glass Cluster and Replay Attack Datasets

Attack	Classifier	Acc(%)	TPR(%)	FPR(%)	FNR(%)
Cluster	RF	78.36	79.3	22.5	20.7
	MLP	72.45	75.1	30.0	24.9
	RT	76.34	72.0	19.6	28.0
Replay	RF	77.86	81.8	26.0	18.2
	MLP	62.76	67.7	42.2	32.3
	RT	74.74	78.1	28.6	21.9

Random Forest achieves best accuracy for both cluster and replay attacks. Movee is more accurate on Glass than on smartphone.

Figure 3. Movee performance (when using various supervised learning algorithms) on several attack datasets built using data collected from 13 participants, (left) on Samsung smartphone, (b) on Google Glass device.

We have introduced novel attacks that focus on Movee's defenses, to fabricate acceleration data that mimics the motion observed in targeted videos. We have used smartphones and wearable smart glasses to collect both genuine and attack data from 13 users. Our experiments show that Movee is able to efficiently detect human and automatically generated plagiarized videos: Movee's accuracy ranges between 68-93 percent on a smartphone, and between 76-91 percent on a Google Glass device.

1.2. Vamos: Video Accreditation Through Motion Signatures

We designed Movee to work on 6s long video and acceleration samples, requiring the user to pan the camera in a specific direction rather than gracefully accept the natural motion of the user. Furthermore, Movee is vulnerable to potent attacks. For example, an attacker starts Movee and points to a portion of a target video playing on a projection screen, performs a pan motion as specified by Movee, then points the camera to the whole frame of the fraudulent video. Since Movee only uses the initial 6s chunk, the resulting sample passes Movee's verifications. These limitations significantly impact the practical application of Movee. To address these limitations, we have designed Vamos,

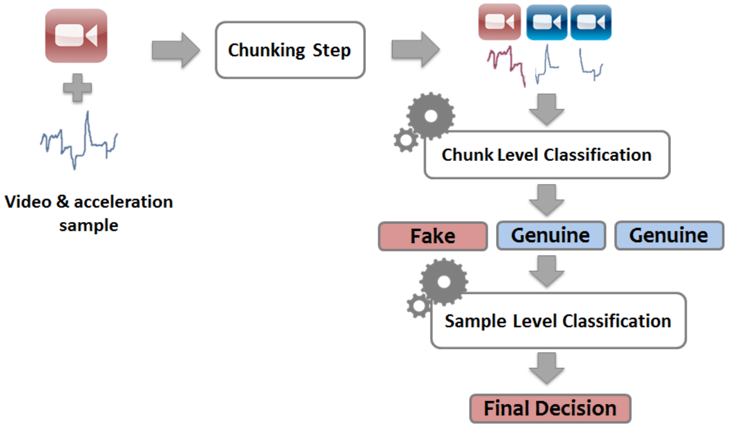

Video & acceleration sample
Chunking Step
Chunk Level Classification
Fake
Genuine
Genuine
Sample Level Classification
Final Decision

Figure 4. Illustration of the Vamos architecture and operation. Vamos consists of three steps, (i) "chunking", to divide the (video, acceleration) sample, (ii) chunk level classification, and (iii) sample level classification.

We introduce Vamos (Video Accreditation Through Motion Signatures) to address these limitations and provide the first video liveness verification system that works on unconstrained, free-form videos, does not impose a "verification" step on users, and is resilient to a suite of powerful, sensor based attacks. The verifications of Vamos leverage the entire video and acceleration sample. This is in contrast with Movee, that relies only on the initial section of the sample. Vamos consists of the process illustrated in Figure 4. First, it divides the input sample into equal length chunks. Second, it classifies each chunk as either genuine or fraudulent. Third, it combines the results of the second step with a suite of novel features to produce a final decision for the original sample.

We introduce several chunking techniques, including sequential, segment based and random. We adapt Movee to provide an improved chunk-level classification: we use DTW, stretching and calibration features for the projection of video and acceleration streams on each axis, instead of for the overall samples. We introduce a suite of novel features to classify a whole samples as genuine or fraudulent; the features are based on the output of the chunk level classification process and on aggregates computed over the chunks.

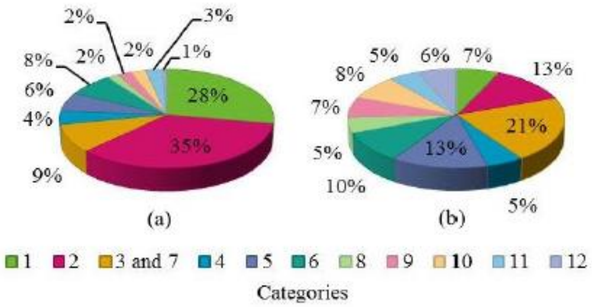
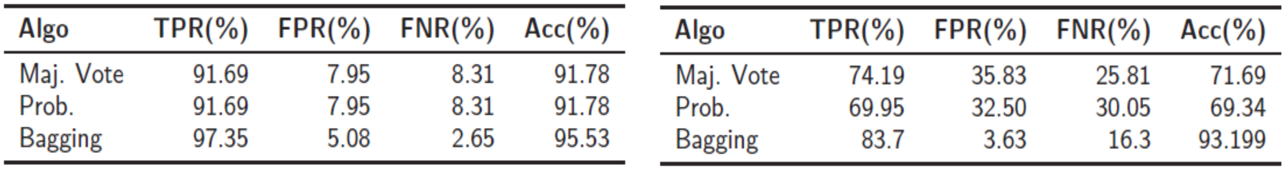


Figure 5. (a) Motion category distribution for YouTube dataset. (b) Distribution for free-form dataset. Table 1 defines the 12 categories.

We have introduced several novel and powerful attacks, including manual "sandwich" attacks, automatic "clustering" attacks, and hybrid "stitch" attacks. We have proposed a novel classification of mobile videos based on the motion of the user holding the device, the motion of the camera, and the distance between the camera and the subject (12 categories). We have performed a user study with 16 participants from whom we have collected 160 free form video and acceleration samples of 30s each. We have used this data and these participants to create datasets of attack data for each attack that we proposed. We have collected 150 citizen journalism videos from YouTube (witnessing armed conflicts from Ukraine, Venezuela, and natural disasters). Figure 5 shows the results of manual labeling of the videos according to our 12 mobile video categories.



Algo	TPR(%)	FPR(%)	FNR(%)	Acc(%)
Maj. Vote	91.69	7.95	8.31	91.78
Prob.	91.69	7.95	8.31	91.78
Bagging	97.35	5.08	2.65	95.53

Algo	TPR(%)	FPR(%)	FNR(%)	Acc(%)
Maj. Vote	74.19	35.83	25.81	71.69
Prob.	69.95	32.50	30.05	69.34
Bagging	83.7	3.63	16.3	93.199

Figure 6. (left) Vamos accuracy on cluster based stitch attack. (right) Vamos accuracy on sandwich based stitch attack. The classifier performs consistently better than manual threshold based alternatives. The accuracy of Vamos exceeds 93% even for the new powerful attacks that we

Figure 6 shows the results of Vamos on the powerful new attacks we introduced. It shows that Vamos achieves an accuracy that exceeds 93%.

2. Securing Mobile Device Storage and Communications

2.1 xRay: In VM Memory Mining for Provenance Tracking

KXRay is designed to detect the existence and location of specific instances of target data structure types in kernel or VM by observing memory accesses and training for target-specific timing-based signatures. The intuition derives from the idea that universal scheduling and process management invariants reflect in access patterns and can be trained for efficiently. Further, at function level, entropy is determined by a limited set of inputs and as a result relative intra-function memory accesses may feature stability and specificity since they are usually directly tied to underlying object types. KXRay can be deployed to defeat kernel rootkits that "hide" their associated processes from existing snapshot-based detection methods.

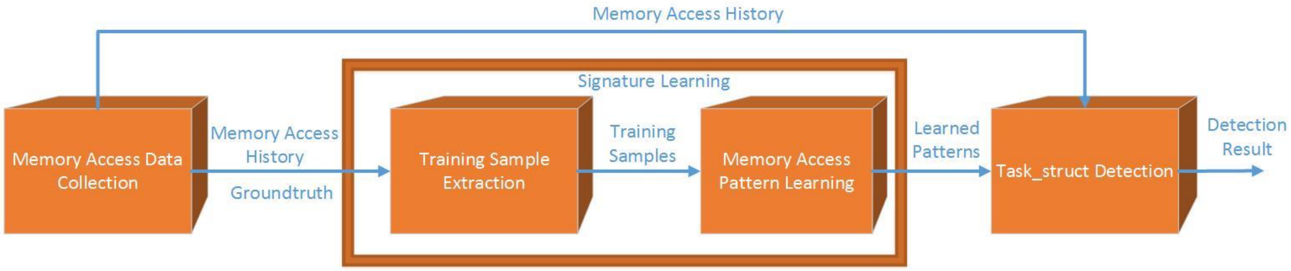


Figure 7. KXRay main framework overview.

We introduce multiple signature variants and evaluate them for different kernel versions. In initial results, trained-for signatures are resilient across the same major kernel version but lose effectiveness for far- removed kernel versions. KXRay successfully detected previously undetected processes hidden by four traditional rootkits. Online analysis of timing access patterns is effective in detecting previously undetectable hidden processes but incurs a heavy performance penalty which will need to be mitigated by specialized processor support or inter-core monitoring software to be acceptable in production.

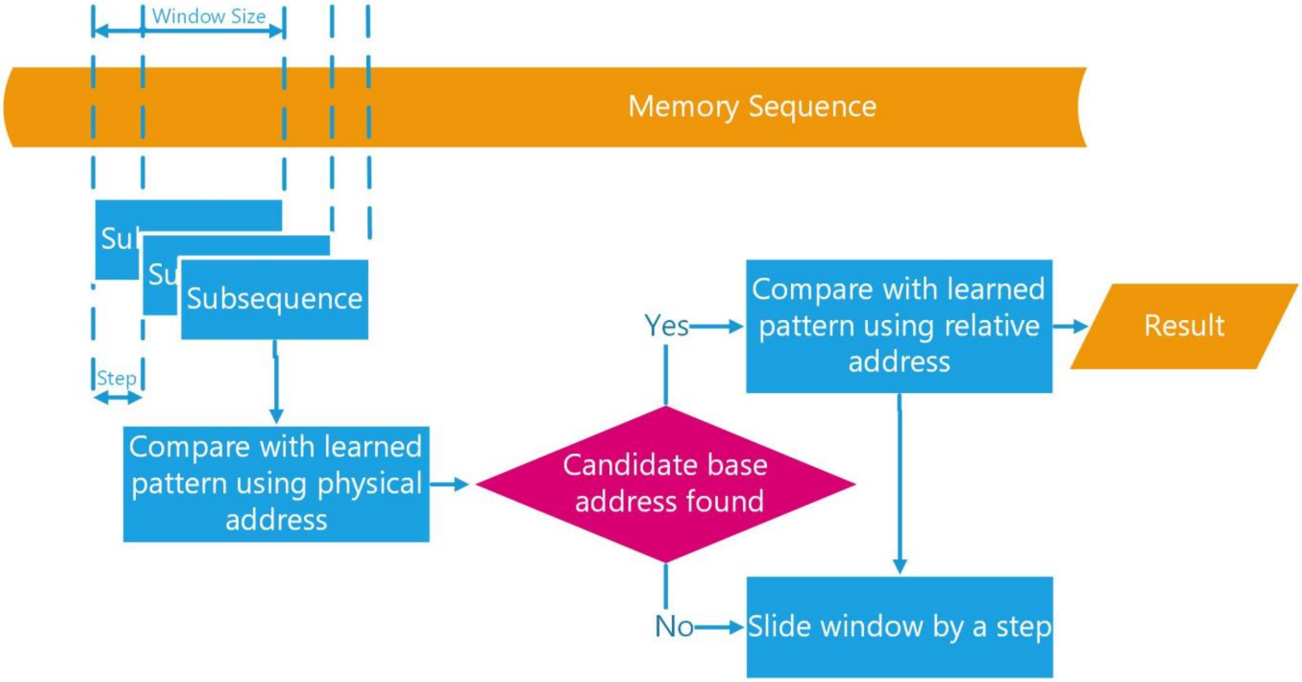

Window Size
Memory Sequence
Su
Su
Subsequence
Step
Compare with learned pattern using physical address
Candidate base address found
Yes
No
Compare with learned pattern using relative address
Result
Slide window by a step

Figure 8. KXRay detection mechanism details.

TrustZone provides one out of band such monitoring mechanism that we can leverage and deploy inside the Secure World to detect malware attacking the normal world kernel and applications and thus protect any sensing code and data streams running in the normal world.

2.2 DroidShield

We have built DroidShield, a system that provides a new Android/TrustZone protection paradigm that enables MCloud to protect user-land application data from all unauthorized accesses, even those originating from a compromised kernel, with the highest privilege. MCloud context data gathered by smartphone sensors can now be relayed correctly and with integrity to its intended trusted MCloud code.

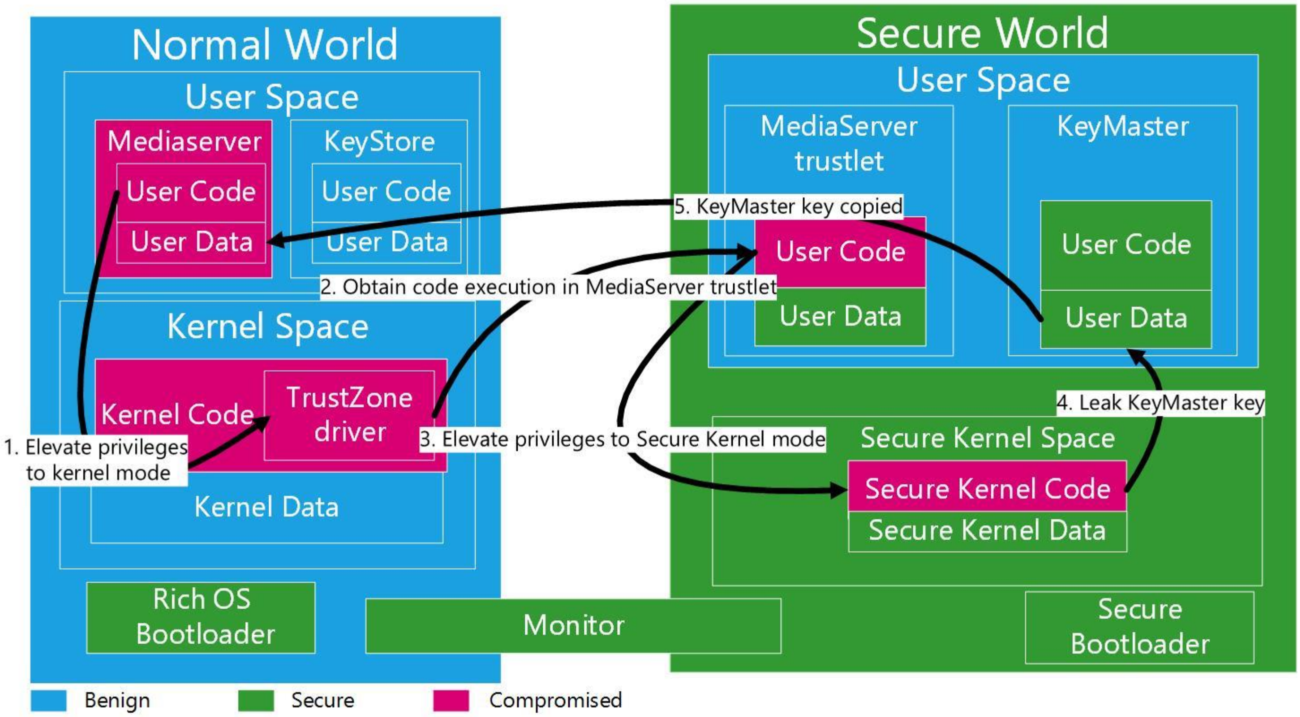


Figure 9. Simple attack example of existing Android malware that leaks data even from code running in the secure world.

DroidShield, provides protection against attacks such as these. Its main goals are to provide a number of strong security properties for logic running on a protected smartphone including

- **For user-land applications**
  - Data confidentiality & integrity
  - Code integrity
  - Secure I/O communication
- **For Normal World Kernel**
  - Code Integrity
  - Process security data integrity
- **For logic running in the Secure World**

- Code & data confidentiality
- Code & data integrity
- Isolation from Normal World access
- No Secure World applications
- No Secure World I/O drivers

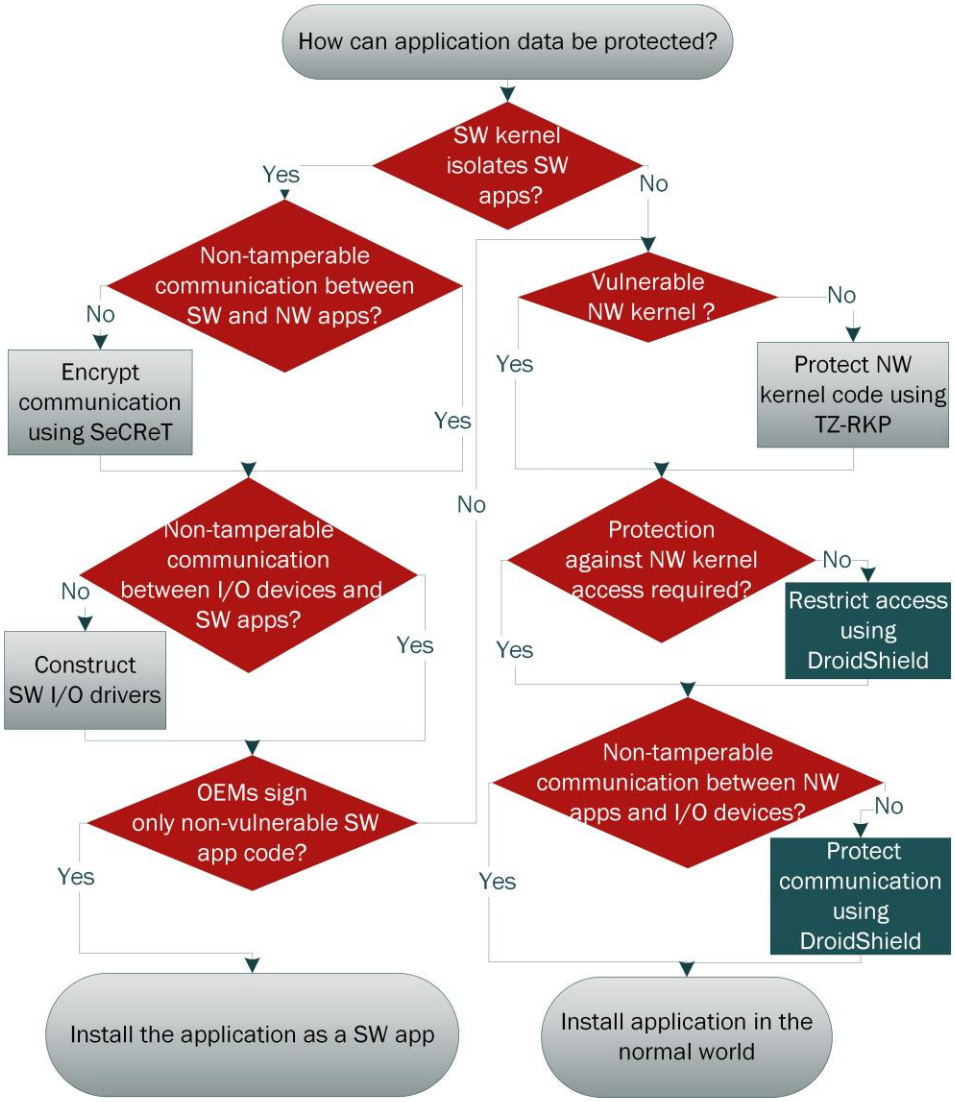

How can application data be protected?
SW kernel isolates SW apps?
Yes
No
Non-tamperable communication between SW and NW apps?
No
Encrypt communication using SeCReT
Yes
Vulnerable NW kernel ?
No
Protect NW kernel code using TZ-RKP
Yes
No
Non-tamperable communication between I/O devices and SW apps?
No
Construct SW I/O drivers
Yes
Protection against NW kernel access required?
No
Restrict access using DroidShield
Yes
OEMs sign only non-vulnerable SW app code?
Yes
Non-tamperable communication between NW apps and I/O devices?
No
Protect communication using DroidShield
Yes
Install the application as a SW app
Install application in the normal world

Figure 10. In effect DroidShield addresses the remaining weak spots in existing application data protection mechanisms to ensure user-land hosted and driven sensor data collection logic is not compromised.

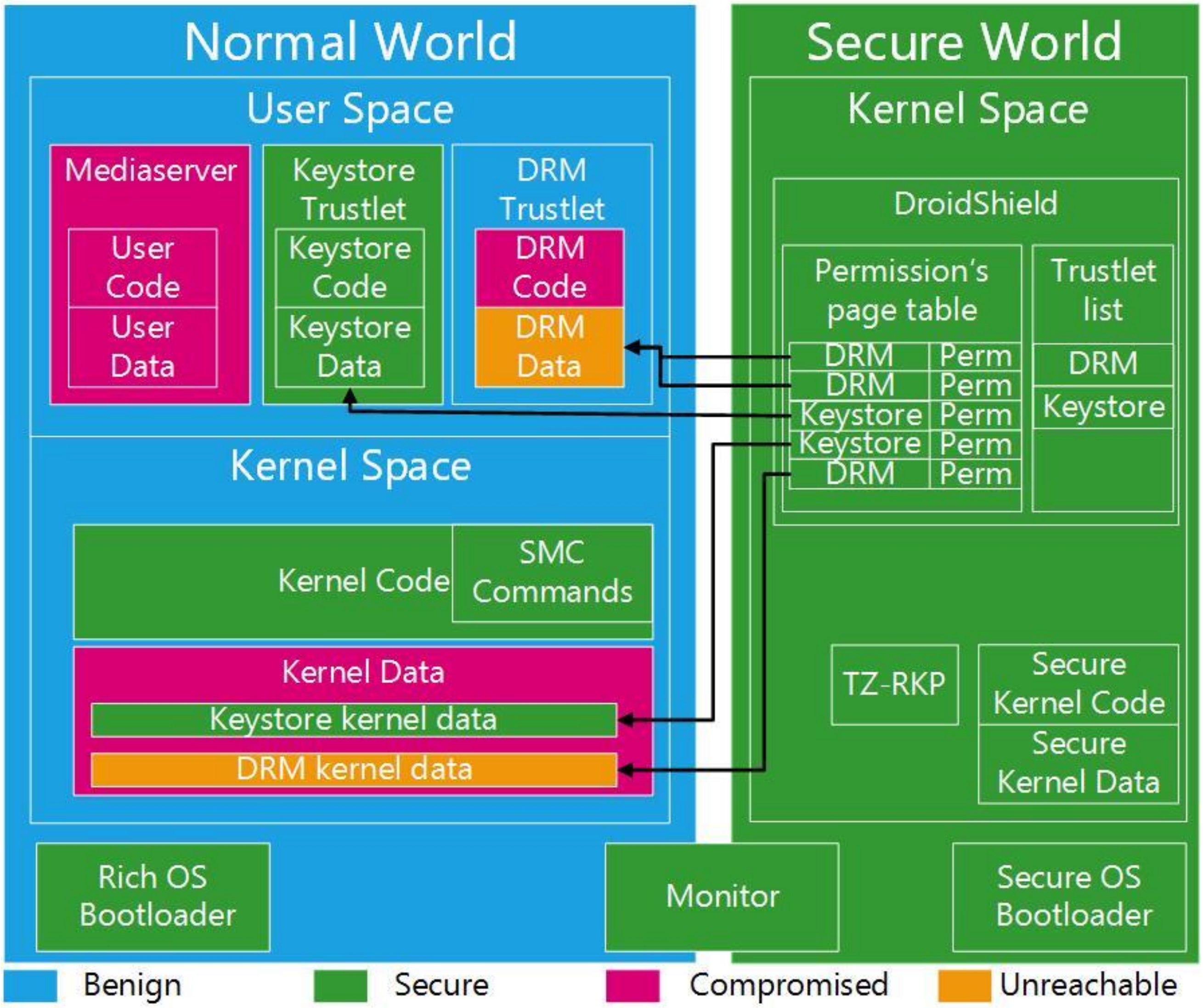


Figure 11. DroidShield Architecture Overview.

DroidShield Solution Summary:

- Protect application memory pages using Secure World
- Isolate Secure World from both I/O and Normal World access
- Minimize Secure World Trusted Computing Base
- No vulnerable OEM code in Secure World, that can compromise its security

2.1. SensCrypt

We have reverse engineered and identified security vulnerabilities in Fitbit and Garmin, two popular and representative fitness tracker products. We have built two attack tools, FitBite and GarMax, and showed how they inspect and inject data into nearby Fitbit Ultra and Garmin Forerunner trackers, see Figure 12 for an

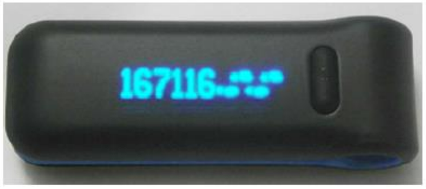


Figure 12. Illustration of injection attack on Fitbit Ultra.

illustration. The attacks are fast, thus practical even during brief encounters. We believe that, the vulnerabilities that we identified in the security of Fitbit and Garmin are due to the many constraints faced by solution providers, including time to release, cost of hardware, battery life, features, mobility, usability, and utility to end user. Unfortunately, such a constrained design process often puts security in the back seat.

We have devised SensCrypt (see Figure 13) a protocol for secure data storage and communication, for use by makers of affordable and lightweight sensors. SensCrypt thwarts not only the attacks we introduced, but also defends against powerful JTAG Read attacks. Thus SensCrypt provides defenses against an attacker that can intercept and modify the communications of sensors or that can even physically capture and access the memory of sensors. SensCrypt leverages the intermittent connectivity of sensors to the Internet in order to reset their memory with pseudo-random one time pads. Data captured by the sensor is then first encrypted with a device key before being xor-ed with the one time pad, into the sensor memory. Thus, in order to recover sensor data, an attacker needs to not only capture its communications, but also physically capture the device twice: once before sensor data is written on its memory, and once after.

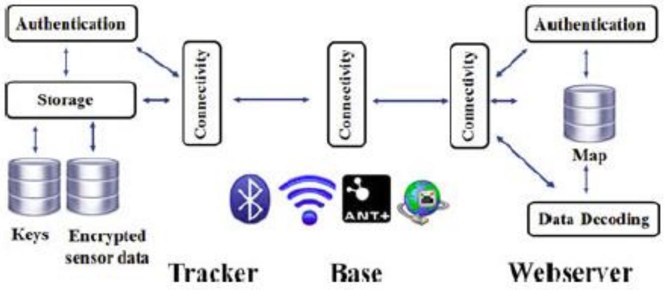


Figure 13. Illustration of SensCrypt Architecture. The sensor stores key material and encrypted data, which can only be accessed by an authenticated server.

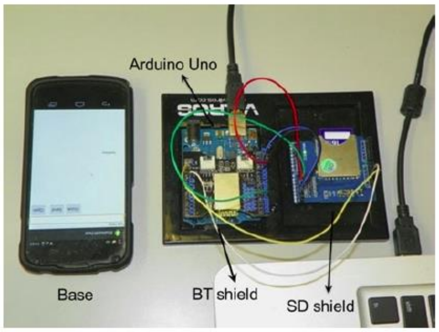


Figure 14. Testbed for SensCrypt. Sens.io is the Arduino Uno device equipped with Bluetooth shield and SD card is the tracker. Nexus 4 is the base.

We have built Sens.io, a prototype tracker, from off-the shelves components, see Figure 14. It consists of an Arduino Uno Rev3 and external Bluetooth (Seeeduino V3.0) and SanDisk card shields. The Arduino platform is a good model of resource constrained trackers: its ATmega328 micro-controller has a 16 MHz clock, 32 KB Flash memory, 2 KB SRAM and 1 KB EEPROM. The Bluetooth card has a default baud rate of 38,400 and communication range up to 10 m. Since the Arduino has 2 KB SRAM, it can only rely on 1,822 bytes to buffer data for transmissions. The SD card (FAT 16) can be accessed at the granularity of 512 byte blocks. The cost of Sens.io is $52 ($25 Arduino card, $20 Bluetooth shield, $2.5 SD Card shield, $4 SD card, see Fig. 9), a fraction of Fitbit's ($99) and Garmin's ($299) trackers.

We have shown that on Sens.io, SensCrypt (i) imposes a 6 ms overhead on tracker writes, (ii) reduces the end-to-end overhead of data uploads to 50 percent of that of Fitbit, and (iii) enables a server to support large volumes of tracker communications.

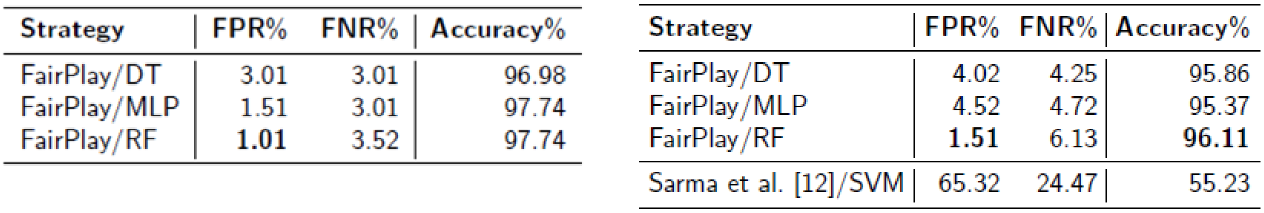
3. Fraud Detection in Social Media

We have developed solutions to identify fraudulent behaviors in Yelp and Google Play. In the following, we summarize our results for each system.

3.1 FairPlay: Detection of Search Rank Fraud and Malware Apps in Google Play

We have performed a detailed temporal analysis of Google Play, Google's app market, on data that we collected daily from 160,000 apps, over a period of six months in 2012. We have discovered that at most 50% of the apps are updated in all categories, which significantly impacts the median price. The average price does not exhibit seasonal monthly trends and a changing price does not show any observable correlation with the download count. We have also shown that productive developers are not creating many popular apps, but a few developers control apps which dominate the total number of downloads. In addition, we have collected longitudinal app data from 87,000 apps, 2.9 million reviews, and 2.4 million reviewers, over half a year, between 2014 and 2015.

We have leveraged this data to develop FairPlay, a novel system that uncovers both malware and search rank fraud apps, by picking out trails that fraudsters leave behind. To identify suspicious apps, FairPlay's PCF algorithm correlates review activities and uniquely combines detected review relation with linguistic and behavioral signals gleaned from longitudinal Google Play app data. For instance, the high cost of setting up valid Google Play accounts forces fraudsters to reuse their accounts across review writing jobs, making them likely to review more apps in common than regular users. Resource constraints can compel fraudsters to post reviews within short time intervals. Legitimate users affected by malware may report unpleasant experiences in their reviews. Ramps in the number of "dangerous" permissions requested by apps may indicate benign to malware (Jekyll-Hyde) transitions.

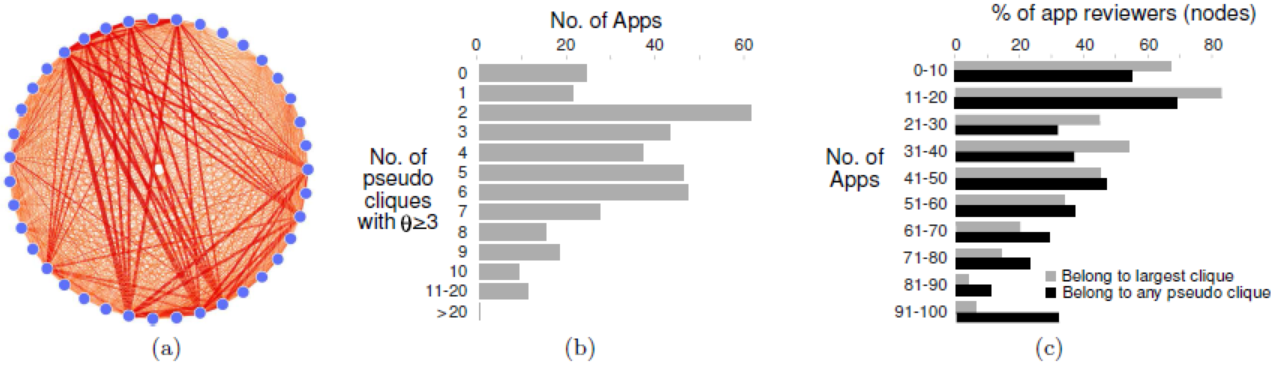


Strategy	FPR%	FNR%	Accuracy%
FairPlay/DT	3.01	3.01	96.98
FairPlay/MLP	1.51	3.01	97.74
FairPlay/RF	**1.01**	3.52	97.74

Strategy	FPR%	FNR%	Accuracy%
FairPlay/DT	4.02	4.25	95.86
FairPlay/MLP	4.52	4.72	95.37
FairPlay/RF	**1.51**	6.13	**96.11**
Sarma et al. [12]/SVM	65.32	24.47	55.23

Figure 15. (left) FairPlay classification results (10-fold cross validation) of gold standard fraudulent (positive) and benign apps. RF has lowest FPR, thus desirable. (right) FairPlay classification results (10-fold cross validation) of gold standard malware (positive) and benign apps, significantly outperforming Sarma et al. FairPlay's RF achieves 96.11% accuracy at 1.51% FPR.

We contributed a longitudinal dataset of 87, 223 freshly posted Google Play apps (along with their 2.9 million reviews, from 2.3 million reviewers) collected between October 2014 and May 2015. We have leveraged search rank fraud expert contacts in Freelancer, anti-virus tools and manual verifications to collect gold standard datasets of hundreds of fraudulent, malware and benign apps.



(a) (b) (c)

Figure 16. (a) Clique flagged by FairPlay for “Tiempo - Clima gratis”, one of 201 seed fraud apps we identified, involving 37 reviewers (names hidden for privacy); edge weights proportional to numbers of apps reviewed in common (ranging from 115 to 164 apps). (b & c) Statistics over the 372 fraudulent apps detected by FairPlay, out of 1, 600 investigated: (b) Distribution of per app number of discovered pseudo cliques. 93.3% of the 372 apps have at least 1 pseudo clique of size 3. (c) Distribution of percentage of app reviewers (nodes) that belong to the largest pseudo clique and to any clique. 8% of the 372 apps have more than 90% of their reviewers involved in a clique!

We have shown that FairPlay achieves high accuracy in differentiating between fraudulent and benign apps as well as between malware and benign apps, see Figure 15. We have shown that 75% of the identified malware apps engage in search rank fraud. FairPlay discovers hundreds of fraudulent apps that currently evade Google Bouncer’s detection technology, see Figure 16 for an illustration of those apps.

3.2 Marco: Fraud Detection in Yelp

While malicious behaviors may occasionally be performed by inexperienced fraudsters, they may also be professionally organized. For example, search engine optimization (SEO) companies tap into review writer markets to offer review campaigns or “face lift” operations for business owners, to manipulate venues’ ratings (1–5 star) through multiple, coordinated artificial reviews. For business owners, profit seems to be the main incentive to drive them to engage in deceptive activities. Studies have shown that an extra half-star rating on Yelp causes a restaurant to sell out 19% more often, and a one-star increase leads to a 5–9% increase in revenue.

City	Car Shop	Mover	Spa
Miami, FL	1000 (6)	348 (8)	1000 (21)
San Fran., CA	612 (59)	475 (45)	1000 (42)
NYC, NY	1000 (8)	1000 (27)	1000 (28)

Figure 17. Venues identified by Marco as fraudulent (shown in red) out of venues of specific type in 3 major US cities.

We introduced Marco (MAlicious Review Campaign Observer), a novel system that leverages the wealth of spatial, temporal and social information provided by Yelp, to detect venues that are targets of deceptive behaviors. Marco exploits fundamental fraudster limitations to identify venues with (i) abnormal review spikes, (ii) series of dissenting reviews and (iii) impactful but suspicious reviews. Marco detects both venues that receive large numbers of fraudulent reviews, and venues that have insufficient genuine reviews to neutralize the effects of even small scale campaigns. The table in Figure 17, shows the ability of Marco to identify real life fraud (numbers shown in red) among venues of various types in different cities.

3.3 Friend Spam Detection with Privacy

Friend spam, adversarial invitations sent to social network users, exposes victims to a suite of privacy, spear phishing and malware vulnerabilities. In this project we have proposed to use the location history of users to detect friend spam. We posited that the user trust in friends is associated with their co-location frequency. To evaluate this hypothesis, we performed a user study on 68 participants. Figure 18 summarizes our findings that the participants tended to be closer friends with, and have more involved topics of conversation with the Facebook friends with whom they meet more frequently.

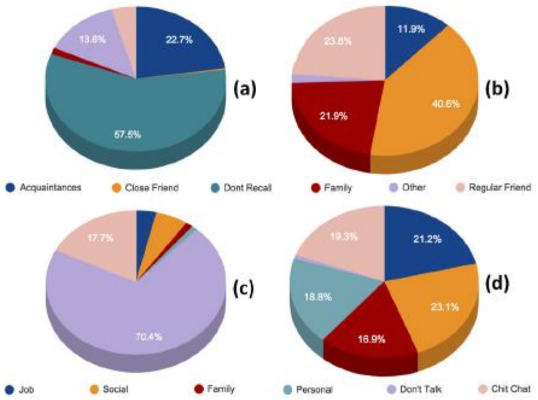


Figure18. Distribution of types of friends for (a) friend never met in person and (b) friends met daily or weekly, and of the topics of discussion for (c) friends never met in person and (d) friends met daily or weekly.

We have leveraged this result to introduce and build GeoPal, a framework that carefully accesses the potentially sensitive location history of users to privately prove their past location claims, and to privately compute and update fuzzy co-location affinities with other users. Figure 19 illustrates GeoPal: mobile devices engage in protocols with venues visited to privately retrieve ``tokens'', proofs of presence in a time and space point. The history of tokens is then used to engage in privacy preserving protocols with prospective Facebook friends, in order to prove trust. We have built GeoPal on PLP, a protocol we developed to privately collect proofs of user past locations. We have shown that GeoPal is practical: a Nexus 5 can process more thank 20K location proofs per second.

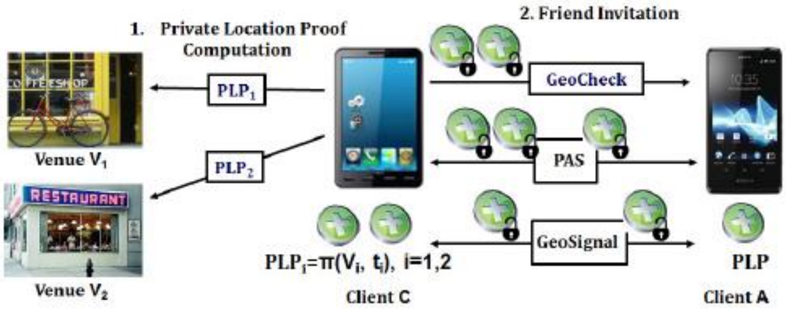


Figure 17. GeoPal architecture. The user's mobile device privately captures and stores proofs of locations visited. The collected proof history is stored on the device and used to process friend invitations: privately prove past locations, determine fuzzy co-location affinity with invited friends, and detect current co-location with pending friends.